LAB EXERCISE

Table Names: Customers, Products, Orders, Order_Details

A. Using the IN Condition

The **SQL IN** clause allows you to specify discrete values in your SQL WHERE search criteria.

THE **SQL IN** syntax looks like this:

```
SELECT Column1, Column2, Column3, …
FROM Table1
WHERE Column1 IN (Valu1, Value2, …)
```

Lets use the EmployeeHours table to illustrate how **SQL IN** works:

Employee	**Date**	**Hours**
John Smith	5/6/2004	8
Allan Babel	5/6/2004	8
Tina Crown	5/6/2004	8
John Smith	5/7/2004	9
Allan Babel	5/7/2004	8
Tina Crown	5/7/2004	10
John Smith	5/8/2004	8
Allan Babel	5/8/2004	8
Tina Crown	5/8/2004	9

Consider the following SQL query using the **SQL IN** clause:

```
```

```
SELECT *
FROM EmployeeHours
WHERE Date IN ('5/6/2004', '5/7/2004')
```

This SQL expression will select only the entries where the column Date has value of '5/6/2004' or '5/7/2004', and you can see the result below:

Employee	Date	Hours
John Smith	5/6/2004	8
Allan Babel	5/6/2004	8
Tina Crown	5/6/2004	8
John Smith	5/7/2004	9
Allan Babel	5/7/2004	8
Tina Crown	5/7/2004	10

We can use the **SQL IN** statement with another column in our EmployeeHours table:

```
SELECT *
FROM EmployeeHours
WHERE Hours IN (9, 10)
```

The result of the SQL query above will be:

Employee	Date	Hours
John Smith	5/7/2004	9
Tina Crown	5/7/2004	10
Tina Crown	5/8/2004	9

What is the use of the IN condition?

IN statement is used to test a set of values

1. List down the products which are from group A and C?
   SELECT DESCRIPTION, PROD_GROUP
   FROM PRODUCTS
   WHERE PROD_GROUP IN ('A','C');

DESCRIPTION	P
Medium Cattle Trough	A
Large Cattle Trough	A
Bicycle Stand	C
Football Player Statue	C

2. List down the products which are from group A, C and K?
   SELECT DESCRIPTION, PROD_GROUP
   FROM PRODUCTS
   WHERE PROD_GROUP IN ('A','C','K');

DESCRIPTION	P
Medium Cattle Trough	A
Large Cattle Trough	A
Bicycle Stand	C
Football Player Statue	C

3. Who are the customers from town Mill Park and Bundoora from customer table?
   SELECT CUST_NAME, TOWN
   FROM CUSTOMERS
   WHERE TOWN IN ('bundoora', 'Mill Park');

CUST_NAME	TOWN
Nevs Nursery	bundoora
Glads Gladdies	Mill Park
Mill Park	Mill Park

4. Look for customers having a current limit of RM1000 and RM3000?

SELECT CUST_NAME, CURR_BALANCE
FROM CUSTOMERS
WHERE CURR_BALANCE IN (3000,4000);

CUST_NAME	CURR_BALANCE
Preston City	4000

EXERCISE : Using the LIKE Condition

LIKE condition used to search for string values with a given condition:

1. Who are the customers having the customer code starting with ‘2’ from customer table?
   SELECT CUST_NAME, CUST_NO
   FROM CUSTOMERS
   WHERE CUST_NO LIKE '2%';

CUST_NAME	CUST_NO
Glads Gladdies	2001
Mill Park	2002

2. List down the customer name and code from town which start with the letter ‘W’?
   SELECT CUST_NAME, CUST_NO
   FROM CUSTOMERS
   WHERE TOWN LIKE 'W%';

CUST_NAME	CUST_NO
Di Hunter	1776

3. What are the products which have a letter ‘t’ at start of the word, middle and ending?
   SELECT PROD_COD, DESCRIPTION
   FROM PRODUCTS
   WHERE DESCRIPTION LIKE 't%'
   OR
   DESCRIPTION LIKE '%t%'
   OR
   DESCRIPTION LIKE 't%';

PROD_COD	DESCRIPTION
MOO	Medium Cattle Trough
LOO	Large Cattle Trough
STANK	Small Septic Tank
LTANK	Large Septic Tank
STAND	Bicycle Stand
GABBY	Football Player Statue

4. Who are the customers having their name ending with ‘r’?
   SELECT CUST_NAME
   FROM CUSTOMERS
   WHERE CUST_NAME LIKE '%r';

CUST_NAME
Di Hunter

B. SQL AND & OR OPERATOR

The **SQL AND** clause is used when you want to specify more than one condition in your SQL WHERE clause, and at the same time you want all conditions to be true. For example if you want to select all customers with FirstName "John" and LastName "Smith", you will use the following SQL expression:

```
SELECT * FROM Customers
WHERE FirstName = 'John' AND LastName = 'Smith'
```

The result of the SQL query above is:

FirstName	LastName	Email	DOB	Phone
John	Smith	John.Smith@yahoo.com	2/4/1968	626 222-2222

The following row in our Customer table, satisfies the second of the conditions (LastName = 'Smith'), but not the first one (FirstName = 'John'), and that's why it's not returned by our SQL query:

FirstName	LastName	Email	DOB	Phone
James	Smith	jim@supergig.co.uk	20/10/1980	416 323-8888

The **SQL OR** statement is used in similar fashion and the major difference compared to the SQL AND is that OR clause will return all rows satisfying any of the conditions listed in the WHERE clause.

If we want to select all customers having FirstName 'James' or FirstName 'Paula' we need to use the following SQL statement:

```
SELECT * FROM Customers
WHERE FirstName = 'James' OR FirstName = 'Paula'
```

The result of this query will be the following:

FirstName	LastName	Email	DOB	Phone
Paula	Brown	pb@herowndomain.org	5/24/1978	416 323-3232
James	Smith	jim@supergig.co.uk	20/10/1980	416 323-8888

You can combine AND and OR clauses anyway you want and you can use parentheses to define your logical expressions.
Here is an example of such a SQL query, selecting all customers with LastName 'Brown' and FirstName either 'James' or 'Paula':

```
SELECT * FROM Customers
WHERE (FirstName = 'James' OR FirstName = 'Paula') AND LastName = 'Brown'
```

The result of the SQL expression above will be:

FirstName	LastName	Email	DOB	Phone
Paula	Brown	pb@herowndomain.org	5/24/1978	416 323-3232

EXERCISE

AND Operator:

1. Identify the customers staying in Mill Park and having a current limit of RM1000?
   SELECT CUST_NAME, TOWN, CR_LIMIT
   FROM CUSTOMERS
   WHERE CR_LIMIT =1000 AND TOWN ='Mill Park';
   no rows selected

2. Identify the customers name starting with the letter 'N' with the postcode 3083?
   SELECT CUST_NAME
   FROM CUSTOMERS
   WHERE CUST_NAME LIKE 'D%'
   AND
   POSTCODE = 3083;
   no rows selected

3. From the order_details table, identify the products which have a order quantity of more than 5 and with an order price of $45.
   SELECT ORDER_NO, ORDER_QTY, ORDER_PRICE
   FROM ORDER_DETAILS
   WHERE ORDER_QTY > 5
   AND
   ORDER_PRICE = 45;

ORDER_NO	ORDER_QTY	ORDER_PRICE
1	10	45

4. Create a select statement with AND operator to show customer number
   SELECT CUST_NO
   FROM CUSTOMERS
   WHERE POSTCODE LIKE '3%'
   AND
   CR_LIMIT > 100 ;

CUST_NO
1066
13144
1776
2001
2002

OR Operator

1. Identify products with product group of A or D?
   SELECT DESCRIPTION, PROD_GROUP
   FROM PRODUCTS
   WHERE PROD_GROUP ='A'
   OR
   PROD_GROUP ='D';

DESCRIPTION	P
Medium Cattle Trough	A
Large Cattle Trough	A
Small Septic Tank	D
Large Septic Tank	D
Laundry Tub	D
Garden Gnome	D

2. List down the products which are from product group D or remake level is less than 10 from products table.
   SELECT DESCRIPTION, PROD_GROUP, REMAKE_LEVEL
   FROM PRODUCTS
   WHERE PROD_GROUP ='D'
   OR
   REMAKE_LEVEL < 10;

DESCRIPTION	P	REMAKE_LEVEL
Medium Cattle Trough	A	3
Large Cattle Trough	A	1
Small Septic Tank	D	5
Large Septic Tank	D	2
Laundry Tub	D	15
Garden Gnome	D	150

3. Identify the order which have been made after 1st July 1993 or cust_no starts with number '3'?
   SELECT *
   FROM ORDERS
   WHERE ORDER_DATE > '01-JUL-93'
   OR
   CUST_NO LIKE '3%';

ORDER_NO	ORDER_DAT	CUST_NO
2	02-JUL-93	13144
3	02-JUL-93	1066

4. Create an OR query from table customers about current limit and current balance?
   SELECT CR_LIMIT, CURR_BALANCE
   FROM CUSTOMERS
   WHERE POSTCODE LIKE '3%'
   AND
   CR_LIMIT > 100 ;.

CR_LIMIT	CURR_BALANCE
300	1800
2800	4000
300	2000
400	0
900	1200

NOT Operator

1. List down the customers who are not from Mill Park and Preston.
   SELECT CUST_NAME, TOWN
   FROM CUSTOMERS
   WHERE TOWN NOT IN ('Mill Park', 'Preston');

CUST_NAME	TOWN
Nevs Nursery	bundoora
Di Hunter	Whittlesea

2. List down the customer name and number who do not have a current balance between 100 and 800.

   SELECT *
   FROM CUSTOMERS
   WHERE CURR_BALANCE NOT BETWEEN 100 AND 800;

CUST_NO	CUST_NAME	STREET	TOWN	POSTCODE	CR_LIMIT	CURR_BALANCE
1066	Nevs Nursery	White Hart	bundoora	3083	300	1800
13144	Preston City	High Street	Preston	3072	2800	4000
1776	Di Hunter	Thornton Farm	Whittlesea	3757	300	2000
2001	Glads Gladdies	Childs Road	Mill Park	3082	400	0
2002	Mill Park	Betula Ave	Mill Park	3082	900	1200

3. Show the product codes which do not have a letter ‘M’.
   SELECT PROD_COD, DESCRIPTION
   FROM PRODUCTS
   WHERE PROD_COD NOT LIKE '%M%';

PROD_COD	DESCRIPTION
LOO	Large Cattle Trough
STANK	Small Septic Tank
LTANK	Large Septic Tank
LTUB	Laundry Tub
STAND	Bicycle Stand
GABBY	Football Player Statue

4. List down the customers who do not have an empty current balance from customer table.
   SELECT *
   FROM CUSTOMERS
   WHERE CURR_BALANCE NOT IN (0);

CUST_NO	CUST_NAME	STREET	TOWN	POSTCODE	CR_LIMIT	CURR_BALANCE
1066	Nevs Nursery	White Hart	bundoora	3083	300	1800
13144	Preston City	High Street	Preston	3072	2800	4000
1776	Di Hunter	Thornton Farm	Whittlesea	3757	300	2000
2002	Mill Park	Betula Ave	Mill Park	3082	900	1200

1. Select the row if a customer is from Mill Park or Preston and current balance is 1000.

`
```
SELECT CUST_NAME, TOWN, CURR_BALANCE
FROM CUSTOMERS
WHERE TOWN IN ('Mill Park','Preston')
OR
CURR_BALANCE = 1000;
```

CUST_NAME	TOWN	CURR_BALANCE
Preston City	Preston	4000
Glads Gladdies	Mill Park	0
Mill Park	Mill Park	1200

2. Select products which are less than $400 and from product group A or D.

```
SELECT DESCRIPTION, LIST_PRICE, PROD_GROUP
FROM PRODUCTS
WHERE LIST_PRICE < 400
AND
PROD_GROUP IN ('A','D');
```

DESCRIPTION	LIST_PRICE	P
Medium Cattle Trough	150	A
Large Cattle Trough	300	A
Small Septic Tank	300	D
Laundry Tub	100	D
Garden Gnome	10	D

3. Identify the customers who have the first letter ‘G’ and the third letter is ‘a’ and current limit is more than $100.

```
SELECT CUST_NAME, CURR_BALANCE
FROM CUSTOMERS
WHERE CUST_NAME LIKE 'G_a%';
```

CUST_NAME	CURR_BALANCE
Glads Gladdies	0

4. Who are the customers from Mill Park or Preston with a current balance of more than $500.

```
SELECT CUST_NAME, TOWN, CURR_BALANCE
FROM CUSTOMERS
WHERE TOWN IN ('Mill Park','Preston')
AND
CURR_BALANCE > 500;
```

CUST_NAME	TOWN	CURR_BALANCE
Preston City	Preston	4000
Mill Park	Mill Park	1200

C. Select from Multiple Tables

You can select from more than one table at a time. To do this, simply separate each table with a comma. You should also qualify any references to columns by placing the table name in front, separated by a dot.

We have another table called *Occupation*, which contains the individual's occupation.

OccupationId	IndividualId	JobTitle
1	1	Engineer
2	2	Accountant
3	3	Cleaner
4	4	Attorney
5	5	Sales Executive

SQL statement

We will select from both the *Individual* table and the *Occupation* table. We will qualify any column names by prefixing them with its table's name and a dot.

```
SELECT *
FROM Individual, Occupation
WHERE Individual.FirstName = 'Homer'
```

D. SQL ORDER BY

The **SQL ORDER BY** clause comes in handy when you want to sort your SQL result sets by some column(s). For example if you want to select all the persons from the already familiar Customers table and order the result by date of birth, you will use the following statement:

```
SELECT * FROM Customers
```

```
ORDER BY DOB
```

The result of the above SQL expression will be the following:

FirstName	LastName	Email	DOB	Phone
John	Smith	John.Smith@yahoo.com	2/4/1968	626 222-2222
Steven	Goldfish	goldfish@fishhere.net	4/4/1974	323 455-4545
Paula	Brown	pb@herowndomain.org	5/24/1978	416 323-3232
James	Smith	jim@supergig.co.uk	20/10/1980	416 323-8888

As you can see the rows are sorted in ascending order by the DOB column, but what if you want to sort them in descending order? To do that you will have to add the DESC SQL keyword after your **SQL ORDER BY** clause

```
SELECT * FROM Customers
ORDER BY DOB DESC
```

The result of the SQL query above will look like this:

FirstName	LastName	Email	DOB	Phone
James	Smith	jim@supergig.co.uk	20/10/1980	416 323-8888
Paula	Brown	pb@herowndomain.org	5/24/1978	416 323-3232
Steven	Goldfish	goldfish@fishhere.net	4/4/1974	323 455-4545
John	Smith	John.Smith@yahoo.com	2/4/1968	626 222-2222

If you don't specify how to order your rows, alphabetically or reverse, than the result set is ordered alphabetically, hence the following to SQL expressions produce the same result:

```
```

```
SELECT * FROM Customers
ORDER BY DOB

SELECT * FROM Customers
ORDER BY DOB ASC
```

You can sort your result set by more than one column by specifying those columns in the **SQL ORDER BY** list. The following SQL expression will order by DOB and LastName:

```
SELECT * FROM Customers
ORDER BY DOB, LastName
```